

Abstract of CN1299546

A media manager is designed to provide data flow management and other services for the client applications on devices connected together within a network. These devices are preferably connected together in an IEEE 1934-1995 serial bus network. A device control module is generated for each of the available devices to abstract all functions and requirements of the device, including proper control protocols, physical connections and connection functions for the device. Also, this media manager manages the flow and format of the data transmitted among devices in the network. A user accesses the media manager via an interface and inputs the functions to be completed by the devices connected together in the network. If the proper device is available, the media manager controls and manages the accomplishment of the requested task. If the proper device is unavailable but the requested sub-device is available among the multiple devices, the media manager creates a virtual device by using the sub-device in the multiple devices to implement the requested task. When a task is assigned to this proper device, and the sub-device, the media manager judges whether the data to be transmitted needs to be converted from one format into another. If so, the media manager also controls the format conversion during the data transmission operation. The media manager further provides the functions of network enumeration and registry searching for the client applications to find available services, physical devices and virtual devices.

Formatted ... [1]

Deleted: MEDIA MANAGER FOR CONTROLLING FLOW AND FORMAT OF DATA BETWEEN AUTONOMOUS MEDIA DEVICE AND MANAGEMENT DEVICE WITHIN NETWORK ENVIRONMENT¶

Formatted ... [2]

Formatted ... [3]

[19] 中华人民共和国国家知识产权局

[51] Int. Cl⁷

H04L 12/40

H04L 29/08 G06F 13/00

[12] 发明专利申请公开说明书

[21] 申请号 99805779.7

[43] 公开日 2001 年 6 月 13 日

[11] 公开号 CN 1299546A

[22] 申请日 1999.4.29 [21] 申请号 99805779.7

[30] 优先权

[32] 1998.5.8 [33] US [31] 09/075,047

[86] 国际申请 PCT/US99/09490 1999.4.29

[87] 国际公布 WO99/59072 英 1999.11.18

[85] 进入国家阶段日期 2000.11.3

[71] 申请人 索尼电子有限公司

地址 美国新泽西州

[72] 发明人 H·A·鲁德特克 B·费尔曼

S·D·斯姆耶尔斯

[74] 专利代理机构 中国专利代理(香港)有限公司

代理人 吴立明 王忠忠

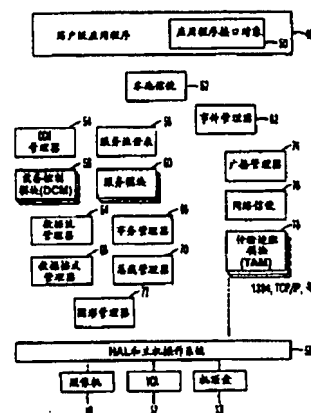
权利要求书 4 页 说明书 21 页 附图页数 6 页

[54] 发明名称 控制网络环境内部的自主媒体设备和管理设备之间的数据的流和格式的媒体管理器

[57] 摘要

一种媒体管理器为网络内连接在设备上的客户应用程序提供数据流管理和其它服务。这些设备最好在 IEEE 1394—1995 串行总线网络内连接在一起。为每个可用设备生成一个设备控制模块,抽象出该设备的全部功能和要求,包括该设备的合适的控制协议、物理连接和连接功能。该媒体管理器也管理网络上设备之间数据传送的流和格式。用户通过一个接口访问媒体管理器并输入要用在网络上连接在一起的设备完成的功能。如果该合适的设备是可用的,媒体管理器控制并管理所请求任务的完成。如果该合适的设备是不可用的,但是所要求的子设备在多个设备中是可用的,媒体管理器就由多个设备中的子设备构成一个虚拟设备,以便完成所请求任务。向该合适的设备和子设备分配给一个任务后,媒体管理器判定是否需要将要发送的数据从一

种格式转换成另一种格式。需要的话,媒体管理器也将数据在数据传送操作期间控制格式转换。媒体管理器也为客户应用程序提供网络枚举和注册表搜索功能,以寻找可用的服务、物理设备和虚拟设备。



ISSN 1008-4274

知识产权出版社出版

权 利 要 求 书

1. 一种管理设备网络的操作和设备网络之间的通信以完成一个任务的方法, 包含的步骤为:

5 a. 确定为完成该任务所需的合适的设备和子设备, 其中, 如果没有合适的设备可用来完成该任务, 由可用的子设备构成虚拟设备; 和

 b. 指令该合适的设备和子设备完成该任务。

10 2. 如权利要求 1 中所要求的方法, 进一步包含为网络中每个设备保持一个控制模块的步骤, 其中, 该控制模块包括该设备和该设备内任何子设备的功能, 并且该控制模块负责该设备的控制。

 3. 如权利要求 2 中所要求的方法, 进一步包含控制该合适的设备和子设备之间的数据流的步骤。

 4. 如权利要求 3 中所要求的方法, 进一步包含的步骤为:

15 a. 获得该网络内各设备的一个拓扑图; 和

 b. 通过分析该拓扑图为该数据流确定一个最佳路线。

 5. 如权利要求 4 中所要求的方法, 进一步包含必要时将在该合适的设备和子设备之间流动的数据转换成适当格式的步骤。

20 6. 如权利要求 5 中所要求的方法, 进一步包含提供一个与要通过其完成该任务的用户相连的接口的步骤。

 7. 如权利要求 6 中所要求的方法, 其中, 该控制模块向一个应用程序提供接口数据并响应来自该应用程序的用户事件。

 8. 如权利要求 7 中所要求的方法, 其中, 该网络是 IEEE 1394 串行总线网络。

25 9. 如权利要求 7 中所要求的方法, 其中, 该控制模块驻留在一个远程设备中, 被下载到一个主设备供执行。

 10. 如权利要求 7 中所要求的方法, 其中, 该控制模块驻留在一个本地设备中, 被从该本地设备内的一个本机环境执行。

30 11. 如权利要求 7 中所要求的方法, 其中, 该控制模块驻留在一个本地设备中, 被上载到一个主设备供执行。

 12. 一种用于控制设备网络的操作和设备网络之间的通信的装置, 包含:

- a. 一个用于与该网络内的设备通信的接口电路; 和
b. 一个与该接口电路相连的控制电路, 用于确定完成该任务所需的合适的设备和子设备和指令该合适的设备和子设备完成该任务, 其中, 如果没有合适的设备可用来完成该任务, 由可用的子设备构成子设备。

13. 如权利要求 12 中所要求的装置, 进一步包含多个各代表该网络中一个设备的控制模块, 其中, 每个控制模块包括一个对应设备和该对应设备内任何子设备的功能, 并且该控制模块负责该设备的控制。

14. 如权利要求 13 中所要求的装置, 进一步包含一个连接到该控制电路以及该接口电路的总线管理器电路, 用于获得该网络内设备的一个拓扑图和通过分析该拓扑图为该数据流确定一个最佳路线。

15. 如权利要求 14 中所要求的装置, 其中, 该控制电路也在有必要进行数据转换时将在该合适的设备和子设备之间流动的数据转换成适当的格式。

16. 如权利要求 15 中所要求的装置, 其中, 该网络是 IEEE 1394 串行总线网络。

17. 一种管理设备网络的操作和设备网络之间的通信的方法, 包含的步骤为:

a. 为网络中每个设备保持一个控制模块, 其中, 该控制模块包括该设备和该设备内任何子设备的能力, 并且该控制模块负责该设备的控制;

b. 提供一个接口, 用户通过它请求一个要完成的任务;

c. 通过搜索控制模块来确定完成该任务所需的合适的设备和子设备; 和

d. 通过指令合适的控制模块向该合适的设备和子设备提供指令而完成该任务。

18. 如权利要求 17 中所要求的方法, 进一步包含的步骤为:

a. 确定该合适的设备和子设备当前是否可用于完成该任务; 和

b. 如果该合适的设备和子设备当前是不可用的, 则由网络内设备的可用的子设备构成虚拟设备完成该任务。

19. 如权利要求 18 中所要求的方法, 进一步包含控制该合适的设备和子设备之间的数据流的步骤。

20. 如权利要求 19 中所要求的方法, 进一步包含的步骤为:

a. 获得该网络内各设备的一个拓扑图; 和

5 b. 通过分析该拓扑图为该数据流确定一个最佳路线。

21. 如权利要求 20 中所要求的方法, 进一步包含的步骤为:

a. 确定是否有必要转换在该合适的设备和子设备之间流动的数据; 和

10 b. 如果数据转换有必要, 则将在该合适的设备和子设备之间流动的数据转换成适当的格式。

22. 如权利要求 21 中所要求的方法, 其中, 该网络是 IEEE 1394 串行总线网络。

23. 一种管理设备网络的操作和设备网络之间的通信的装置, 包含:

15 a. 多个各代表该网络中一个设备的控制模块, 其中, 每个控制模块包括一个对应设备和该对应设备内任何子设备的功能, 并且该控制模块负责该设备的控制。

b. 一个用于与用户通信的接口, 其中, 要完成的任务是由用户通过该接口请求的;

20 c. 一个与该多个控制模块、与该网络和与显示器相连的控制电路, 用于通过搜索控制模块来确定完成该任务所需的合适的设备和子设备, 并通过指令合适的控制模块向该合适的设备和子设备提供指令而完成该任务。

24. 如权利要求 23 中所要求的装置, 其中, 该控制电路也确定该合适的设备和子设备当前是否可用于完成该任务, 如果该合适的设备当前是不可用的, 则由网络内设备的可用的子设备构成虚拟设备完成该任务。

25. 如权利要求 24 中所要求的装置, 其中, 该控制电路进一步控制该网络内设备之间的数据流。

30 26. 如权利要求 25 中所要求的装置, 进一步包含一个与该控制电路相连的总线管理器电路, 用于获得该网络内各设备的一个拓扑图和通过分析该拓扑图为该数据流确定一个最佳路线。

27. 如权利要求 26 中所要求的装置, 其中, 该控制电路也在有必要进行数据转换时将在该合适的设备和子设备之间流动的数据转换成适当的格式。

5 28. 如权利要求 27 中所要求的装置, 其中, 该控制电路执行使用户能利用该网络中设备的基本功能的预定动作。

29. 如权利要求 28 中所要求的装置, 其中, 该控制电路也监测和记录用户行动并创建常规的、用户定义的动作。

30. 如权利要求 27 中所要求的方法, 其中, 该网络是 IEEE 1394 串行总线网络。

说明书

控制网络环境内部的自主媒体设备 和管理设备之间的数据的流和格式的媒体管理器

5

本发明涉及管理网络环境内部的应用和设备的领域。更具体来说，本发明涉及管理网络环境内部设备的操作和设备之间的通信的领域。

10 IEEE 1934-1995 标准“高性能串行总线 1934-1995 标准”，是实现支持异步和等时格式数据传送的廉价高速串行总线体系结构的国际标准。等时数据传送是使得重要实例之间的时间间隔在发送和接收应用程序处有相同的持续时间的实时传送。等时数据传送的理想应用的一个例子是从录像机向电视机的数据传送。录像机记录图像和声音并将数据保存在离散的字节片或数据包中。录像机然后将
15 每个代表在一个有限时段记录的图像和声音的数据包，在该时段期间传送，供电视机进行显示。IEEE 1934-1995 标准的串行总线体系结构提供多个用于应用程序之间的等时数据传送的通道。一个 6 位的通道号与数据一起被广播，以保证由适当的应用程序接收。这允许多个应用程序在总线结构上同时地发送等时数据。异步传送是传统的
20 的数据传送操作，异步传送尽可能快地进行，从一个源向一个目的地传送一定数量的数据。

IEEE 1934-1995 标准为互连数字设备提供了高速的串行总线，由此提供一个通用的 I/O 连接。IEEE 1934-1995 标准为应用程序定义了一种数字接口，应用程序由此无需先将数字数据转换成模拟数
25 据后再在总线上发送。相应地，接收的应用程序将从总线接收数字数据而不是模拟数据，因此无需将模拟数据转换成数字数据。IEEE 1934-1995 标准所要求的电缆与其它用来连接这种设备较粗的电缆相比很细。在总线使用中时可以添加或者从 IEEE 1394-1995 总线去除设备。如果添加或去除了一个设备，总线于是将自动地重新配置自己，从而在现有节点之间发送数据。节点被视为是总线结构上具有
30 独有地址的逻辑实体。每个节点提供一个标识 ROM、一个标准化的控制寄存器集合和其自己的地址空间。

媒体设备配备有网络接口，使它们能变成诸如 IEEE 1934-1995 串行总线网络的网络的一部分。在包含这种自主媒体设备的家庭音像网络中，有可能将一个或多个这种设备在网络中与个人电脑、机顶盒或其它包含微处理器的设备连接在一起。目前缺少可用的接口
5 和控制应用程序来有效地管理这种网络配置内的自主设备的交互作用和操作。所需要的是允许网络配置内的控制设备有效地控制该网络内设备之间的通信和设备的操作的接口。另外还需要允许网络配置内的控制设备使网络内用于完成任务和操作的可用设备的数量最大化的接口。

10 一种媒体管理器为网络内连接在一起的设备上的客户应用程序提供数据流管理和其它服务。这些设备最好在 IEEE 1934-1995 串行总线网络内连接在一起。为每个可用设备生成一个设备控制模块，抽象出该设备的全部功能和要求，包括该设备的合适的控制协议、物理连接和连接能力。该媒体管理器也管理网络上设备之间数据传
15 送的流和格式。用户通过一个接口访问媒体管理器并输入要用在网络上连接在一起的设备完成的功能。如果该合适的设备是可用的，媒体管理器控制并管理所请求任务的完成。如果该合适的设备是不可用的，但是所要求的子设备在多个设备中是可用的，媒体管理器就由多个设备中的子设备构成一个虚拟设备，以便完成所请求任务。
20 向该合适的设备和子设备分配给一个任务后，媒体管理器判定是否需要将要发送的数据从一种格式转换成另一种格式。需要的话，媒体管理器也将控制数据传送操作期间的格式转换。媒体管理器也为客户应用程序提供网络枚举和注册表 (registry) 搜索功能，以寻找可用的服务、物理设备和虚拟设备。

25 图 1 表示一个 IEEE 1934-1995 串行总线网络，包括摄像机、盒式磁带录像机、计算机、电视机和机顶盒；

图 2 表示在每个实现本发明的媒体管理器的设备中存在的硬件系统的框图；

图 3 表示本发明的媒体管理器平台的体系结构的框图；

30 图 4 表示本发明的媒体管理器平台的体系结构的详细框图；

图 5 表示用本发明的媒体管理器进行的设置中和在两个设备间进行数据传送时所涉及的步骤的流程图；

图 6 表示在启动期间客户应用程序遵循的流程图。

5 本发明的媒体管理器为网络内的物理设备提供数据流管理和其它服务。物理设备是由销售商以单独部件的形式出售的产品。物理设备的例子包括电视、盒式磁带录像机、个人电脑、摄像机、CD-ROM 播放机等等。其它许多例子也是人们熟知的，能从商业渠道获得的。每个物理设备包括一些子设备。例如，市场销售的典型的摄像机包括多个实现不同功能的子设备，诸如照相机和视频播放机。

10 这些物理设备最好在 IEEE 1394-1995 串行总线网络内连接在一起。为每个可用的设备和子设备生成一个设备控制模块 (DCM)。每个 DCM 提供对每个设备的全部功能和要求的抽象，包括设备的合适的控制协议、物理连接和连接能力。媒体管理器也管理网络上物理设备之间数据传送操作的流和格式，包括在数据传送操作期间将数据转换成不同的格式。

15 用户通过一个接口访问媒体管理器并输入要用网络上的物理设备完成的功能。如果合适的设备是可用的，没有被另外使用，媒体管理器就控制并管理所请求任务的完成。如果合适的设备是不可用的，媒体管理器就由设备内可用的子设备或部件创建一个虚拟设备，以完成所请求任务。向该合适的物理设备和/或子设备分配给一个任务后，媒体管理器判定是否需要将要发送的数据从源设备的格式转换成接收设备的格式。如果需要转换，媒体管理器也将在控制数据
20 传送操作的同时控制该操作。

25 图 1 表示的示例性系统包括由 IEEE 1394-1995 电缆 15、16 和 18 连接在一起的包括摄像机 10、盒式磁带录像机 12、计算机 14、机顶盒 13 和电视机 11。IEEE 1394-1995 电缆 16 连接摄像机 10 与盒式磁带录像机 12，使摄像机 10 能向盒式磁带录像机 12 发送供记录的数据。IEEE 1394-1995 电缆 18 连接盒式磁带录像机 12 与计算机 14，使盒式磁带录像机 12 能向计算机 14 发送供播放的数据。IEEE 1394-1995 电缆 15 连接机顶盒 13 与计算机 14。机顶盒 13 也由电缆 17 连接到电视 11。

30 图 1 所示的这个配置只是示例性的。显然，音像网络可以包括许多不同的物理部件的组合。这种 IEEE 1394-1995 网络内的物理设备是自主设备，就是说，在如图 1 中所示的，计算机是其中的一个

设备的 IEEE 1394-1995 网络中，计算机与其它设备之间没有真正的“主-从”关系。在许多 IEEE 1394-1995 网络配置中，甚至可能不出现计算机。即使在这种配置中，网络内的设备也完全能够平等地彼此交互作用。

5 图 2 表示在实现本发明的媒体管理器的管理设备中存在的硬件系统的框图。在图 2 中所示的硬件系统中，印刷电路板 20 与用户接口 30 相连。印刷电路板 20 包括一个中央处理单元 (CPU) 22，它与系统存储器 24 相连并通过系统总线 28 与 I/O 总线接口 26 相连。术语“CPU”的使用并非意味着这种系统必定是个通用计算电路。相反，
10 这个电路可以用通用控制器或专用电路来实现。用户接口 30 也连接到系统总线 28。用户接口 30 是特定于子系统的，但最好至少包括红外遥控器和显示器。或者，用户接口 30 也包括用于与该子系统的用户通信的其它 I/O 设备。

15 本发明最佳实施例中，将媒体管理器包括在诸如电视或带显示器的计算机的设备中，以方便与用户顺利地交互。不过显然，本发明的媒体管理器也能在包含提供与用户的接口所必需的部件的任何其它功能设备上实现。为了实现本发明的媒体管理器，它在其中实现的每个部件将包括一个诸如图 2 所示系统的硬件系统。这种设备内的 CPU 22 被用来执行应用程序指令。本发明的媒体管理器于是被
20 用来管理网络内的通信和操作。用户通过在控制设备处提供的接口访问媒体管理器。通过这个接口，用户能监测网络和网络内设备的操作和状态。用户也能通过这个接口控制设备和请求要完成的任务。这些任务的一个例子包括在 VCR 12 上播放录制的磁带和在电视 11 上显示 VCR 12 的输出。本发明的媒体管理器也管理数据传送操作和
25 在各个设备请求的任务。

图 3 表示本发明的媒体管理器平台的体系结构的框图。该体系结构被划分成所谓的上半部分 32 和下半部分 34。下半部分 34 最好包括支持目前最通用的商业操作系统的 IEEE 1394-1995 总线接口和功能。上半部分 32 包括汇集基础的 IEEE 1394-1995 总线支持软件，
30 并增加向客户应用程序、因而向用户提供的许多特征功能和增强功能的部件。上半部分 32 包括：方框 46，它为更高级的 IEEE 1394-1995 总线支持软件提供特定的设计和实现；方框 48，它包括并提供与各

种客户应用程序的接口。下半部分 34 包括方框 40、42 和 44，它们分别为最通用的各操作系统—包括 Windows 95®、Machintosh®和 AperiOS™—提供支持软件。也为诸如 OS9 的任何通用操作系统提供支持软件。下半部分 34 也包括：方框 38，它提供 IEEE 1394-1995 支持软件的公用层；方框 36，它提供与连接到控制设备的其它设备的实际的物理 IEEE 1394-1995 总线接口。

该媒体管理器平台提供一个开发的和灵活的体系结构，以便有效地在网络配置中集成个人电脑和其它自主设备，并有效地管理这些设备之间的必要的数据传送操作。该体系结构的下半部分 34 已经被设计得在最低各层支持基础技术，这使较高的层能支持更通用的模块和功能描述。

图 4 表示本发明的媒体管理器平台的体系结构的更详细的框图。多媒体或用户级应用程序 48 位于该体系结构的顶部，并利用媒体管理器所提供的服务。多媒体应用程序 48 是本发明的媒体管理器的一个应用程序或其它客户程序。媒体管理器内的结构部件管理协议的细节并向应用程序 48 输出一个更简单的程序设计接口。诸如定时、缓冲器管理、总线管理和通信协议等问题被隐藏在这些简单功能接口的背后。应用程序 48 也能访问该体系结构的较低各层，并且当然能够与硬件适配层（HAL）和主机操作系统 58 直接通信。主机操作系统与网络内的其它设备—诸如照相机 10、VCR 12 和机顶盒 13—相连。为说明起见，在这个配置中将媒体管理器在图 1 的计算机系统 14 上实现。

在媒体管理器环境中，应用接口对象 50 作为客户应用程序 48 的代理。提供一个应用程序设计接口，使客户应用程序 48 能利用媒体管理器的具体服务。通过提供应用程序 48 对本地信使（local messenger）52 的访问的应用接口对象 50，也提供对由某些程序设计接口提供的对更详细的或特定的函数的访问。

本地信使 52 是在媒体管理器中集成的消息接发系统的一个部件。这个消息接发系统最好是 AV 信使（AV Messenger）系统。本地信使 52 是在给定节点上所有对象存在于单独的执行空间中时这些对象之间的通信的中枢。本质上来说，本地信使 52 是由主机操作系统提供的应用程序间通信模型。本地信使 52 是软件模块之间传送的所

有消息都要经过的瓶颈。为了取得过程命令能力 (scriptability), 本地信使 52 在所有消息经过时记录它们, 保存一个内部数据库, 内容为所有的消息及它们的相关数据, 包括目的地的地址、参数、响应的地址和 (可选的) 用于基于时间的脚本 (scripting) 的时间戳。

5 服务注册表 59 包含一个对媒体管理器 71 内的所有可寻址实体的引用。这个注册表包含对应每个设备控制模块 (DCM) 56、DCM 管理器 54、数据流管理器 64、事务管理器 66、数据格式管理器 68、总线管理器 70 和图形管理器 72 的一个引用。服务注册表 59 也含有任意数量的服务模块, 这将在下文作说明。服务注册表 59 也含有一个服务注册表数据库, 包含其节点的所有本地对象的引用, 在特定时候也包含远程对象的引用。数据库中的每个条目指向一个可寻址模块并包括附属的属性—有些属性是所有条目共有的, 其它的是某类模块专有的。共有属性包括诸如模块名和本地 ID 之类的事项。模块专有属性因模块的类型而异。条目一旦存在于服务注册表中, 15 就能向该条目添加任意数量的属性。当客户应用程序检索数据库时, 应用程序指定一个要匹配的属性集合, 服务注册表 59 搜索数据库, 寻找并返回与指定的条件吻合的所有条目。如果在搜索过程中找到多个候选项, 服务注册表 59 将向客户应用程序 48 提供一个列表引用 (list reference)。客户应用程序然后就能检查列表中的每个候选项, 20 确定感兴趣的项目。

客户应用程序 48 可能有多个未完成 (outstanding) 检索列表, 每个代表一个不同的检索条件的结果。当客户应用程序 48 因发生某个事件, 诸如总线复位, 此时可能有不同的设备可用—需要更新某个检索列表时, 应用程序 48 就在进行检索调用时将该列表索引传 25 送回服务注册表 59。这就使服务注册表 59 能更新现有的列表对象, 而不是清除掉它后再分配一个新的。

服务模块 60 是能被调用去执行某服务集合的模块。服务模块 60 为客户应用程序进行各种服务, 包括诸如数据格式、传输和控制协议转换等服务。

30 DCM 管理器 54 负责处理其本地节点上的各 DCM 56 或者负责控制设备的网络内的设备。这些责任包括发现、实例化和清除某给定系统可用的所有可能的 DCM 候选的任务。此外, DCM 管理器 54 还与远

程节点上的其它 DCM 管理器（如果有的话）通信，以仲裁网络范围内的设备和子设备的资源分配和管理。

5 DCM 管理器 54 与基础操作系统服务协作，以得到可用设备节点句柄（handle）的原始列表。DCM 管理器 54 也提供一个应用程序设计接口，让客户应用程序 48 能发现网络上设备内的哪些（由各 DCM 56 代表的）子设备或其它服务是可用的。DCM 56 代表一个可供 DCM 管理器 54 分配的设备或子设备。DCM 56 可代表一个物理设备或者一个由不同物理设备的子部件（subpart）构成的虚拟设备。其它可用的服务由各虚拟 DCM 56 代表，这将在下文作说明。可用的 DCM 将是动态的，视 IEEE 1394-1995 串行总线上可用的物理设备而定。

10 对于每个节点来说，DCM 管理器 54 作出足够的工作来确定它应当创建一个 DCM 56。这是为将在本发明的媒体管理器的保护下被管理的所有的媒体相关设备作出的。对于每个媒体相关实体，DCM 管理器 54 生成一个类属的（generic）DCM 56。每个 DCM 56 于是有责任使自己更加特定于设备，如下文将要说明的那样。

15 也可以将制造商提供的特定于设备的 DCM，添加到各 DCM 56 中。特定于设备的 DCM 可以有各种来源，包括设备内的内置只读存储器（ROM）或者其它存储机构，诸如磁盘或磁带的磁头。特定于设备的 DCM 也可以从因特网网站或者通过直接的调制解调器连接下载——如果媒体管理器可利用这些功能，或者由软盘或其它存储介质提供。美国专利申请“一种在设备内添加自描述信息的方法和装置”（申请号-，申请日-）中详细讨论了这些备选方案，特此引用。

20 DCM 管理器 54 负责在合适的时间添加或去除各 DCM 56，并通知客户程序已经添加或去除了各 DCM 56。DCM 管理器 54 也负责在多个 DCM 56 之间协调复杂的服务。这些复杂的服务，诸如复杂操作的命令排队，要求 DCM 管理器 54 与多个 DCM 56 协调，以执行这些操作。

25 各 DCM 56 通过输出一个用于归客户应用程序 48 负责的设备控制的标准化接口，提供一个设备模拟和控制协议抽象服务。由各 DCM 56 为设备控制提供的程序设计接口被划分成公用的 A/V 控制和特定于设备的 A/V 控制。公用的 A/V 控制的命令对几乎所有具有音像功能的设备是通用的。这里包括诸如播放、停止、快进、倒退命令等基本传送控制功能。特定于设备的 A/V 控制命令包括对给定类型的 A/V

设备通用的特征功能—诸如对于具有录制功能的设备来说的录制命令，和特定于一定型号或某组设备的特征功能。特定于设备的功能的信息，既可以利用前文提及的自描述数据结构在内置于设备本身的特定于设备的 DCM 中设置，也可以以软件升级的形式从因特网下载。

5 本发明的媒体管理器采用协议抽象 (protocol abstraction)，这意味着在各模块与应用程序之间的程序设计接口是相同的，不管设备的种类和所使用的控制协议如何。相应地，应用程序将用与它用来使视频系统控制体系结构的 (VISCA-Video System Control
10 Architecture) VCR 进行录制的相同的源代码和消息使 IEEE 1394-1995 VCR 进行录制。这对公用的 A/V 控制命令和特定于类型的控制命令来说是正确的；真正特定于特定设备的特征功能将具有独特的程序设计接口。

DCM 56 是通过其将自描述数据从设备下载并表示给用户的机构。这要求 DCM 56 通过下载和汇集模块并通过一个主应用程序管理
15 该信息向用户的表示，分析该自描述信息。这使用户能通过媒体管理器接口配置和控制网络上设备的众所周知的和特定于设备的功能。美国临时专利申请“描述基于 AV/C 的设备的人类接口特征和功能的方法”（申请号 60/054,199，申请日 1997 年 7 月 30 日）中，
20 描述了用户接口数据的最佳表示和对设备的常规功能的利用，特此引用。

DCM 管理器 54 和各 DCM 56 共同进行对要执行的 AV 命令的命令排队，使 DCM 56 能处理所有的设备特质，诸如为补偿设备的机械延迟而进行预转动的需要。DCM 管理器 54 和各 DCM 56 与媒体管理器的
25 其它部分协作，也能够规定在特定时候和因某些条件而采取的设备控制动作。

各 DCM 56 构成了本发明的媒体管理器的总体体系结构的大部分。DCN 56 提供对构成音像设备的所有各种技术项的抽象，诸如控制协议、物理连接和连接功能。DCM 56 也能被创建得不是代表物理
30 设备，而是代表包含一系列执行特定 AV 操作的功能或服务的虚拟设备。

物理设备和子设备是可以单独地使用的硬件。本发明的媒体管

理器用可以使用的子设备支持虚拟设备向设备网络增加增强的功能。虚拟设备是由各种可用的部件组合的逻辑实体。虚拟设备最好在因要完成一个被请求的任务而需要时自动创建。另一方面,虚拟设备也因从 DCM 管理器请求服务而动态地创建。

5 AV 动作是一个预先定义的动作或活动,诸如“看电视”或“录制电影”,或者是任何由用户定义的涉及通过用各 DCM 56 操纵设备的动作的集合。可以将动作记录下来供用户以后再用。AV 动作应用程序接口是一种模拟方式,模拟用 AV 网络中的设备执行的普通动作,诸如观看录象表演、观看广播表演、复制磁带和听激光唱片。

10 例如,如果 VCR 位于用户家中楼上的卧室,当前正在通过调谐器接收广播并将其显示在卧室中的电视上,则 VCR 内部的走带机构不在使用。如果用户然后希望想在楼下的电视上观看录象,本发明的媒体管理器将允许用户把录像带放入卧室中的 VCR 的走带机构中,在楼下的电视上观看该录像带中的录象。代表该录象的数据从楼上的

15 VCR 被通过 IEEE 1394-1995 网络传输到楼下的电视。这个数据传送操作由控制设备中的媒体管理器控制。类似的功能和虚拟设备是用调谐器、多路输出选择器、放大器、处理器以及其它部件和子设备实现的。于是,用户用媒体管理器来控制设备的操作,能最大限度地利用网络内设备的功能和能力。

20 DCM 管理器 54 不仅跟踪记录正在使用什么设备和子设备,也跟踪记录由当前可用的部件和子部件能创建出什么虚拟设备。DCM 管理器 54 为它所有的本地管理的设备和为它在其上执行的主平台上可用的软件服务做这件事。DCM 管理器 54 也为客户应用程序 48 提供程序设计接口,以查问能在网络上可用资源的基础上创建的虚拟设备,

25 以及当前能执行什么 AV 动作。DCM 管理器 54 也保证虚拟设备在合适的时候被添加到服务注册表 59。

 为各 DCM 56 配备的应用程序设计接口,被设计得使客户应用程序 48 能发现网络中的设备内有哪些特征功能 and 能力可用并因此按需与这些设备协作。这个程序设计接口包括设备控制、设备管理、连接管理和自描述设备实现的管理。每个 DCM 56 有责任通过确定它管理的设备的类型而从一般的 DCM 转换成特定于设备或协议的 DCM。这

30 要求 DCM 检查设备中可能出现的自描述设备数据,分析能得到的任

何其它信息。各 DCM 56 也有责任向其它设备和部件开放被管理设备的自描述设备信息数据, 包括设备图象 (image)、产品名称串和功能描述符。DCM 56 进一步负责为包括诸如命令排队等复杂服务的设备控制提供一致的接口。执行这些命令要求与用于设备控制协议应用的主机操作系统协调, 包括通过协议驱动器和操作系统提供的其它支持机构封装、发送、处理特定于协议的命令和应答。

每个 DCM 56 也监测它控制的设备并向必要的部件和应用程序提供扩展的通知支持 (extended notification support)。设备所生成的所有正常事件, 经过 DCM 56 去往合适的设备, 去往事件管理器 62 和去往所有有兴趣的客户应用程序 48。除了支持 AV/C 设备通知事件外, 许多情形可能在 AV/C 协议中、在给定设备或其它控制协议中都得不到明确的支持。根据设备的能力及其控制和通信协议, 有可能为这类不触发实际事件消息的事件提供扩展的支持。DCM 56 观察设备的这种活动, 向事件管理器 62 投寄事件。

每个 DCM 56 也负责根据正在利用在它们控制下的设备的数据的外部实体和控制它们的实体管理自己。这包括支持资源共享和资源排队。资源排队允许某实体预定一个忙的 DCM, 在该 DCM 56 可用时供其使用。一旦该 DCM 56 可用, 它将通知该实体。各 DCM 56 也最好在环境变化过程中继续存在, 使 DCM 和客户程序支持在线和下线两种状态。这使各 DCM 56 能在设备恢复在线后立即重新建立设备的服务。

DCM56 最好在环境改变起见保持一个表现 (presence) (以允许 DCM 和客户支持在线和离线状态两者。这允许 DCM56 在返回在线时快速重建设备的服务。

在本发明的媒体管理器内, 各 DCM 56 负责控制可用的设备和子设备。各 DCM 56 开放通用的和特定于设备的设备能力。在本发明的另一种形式的实施例中, 每个作为自描述数据一部分的设备有一个内置的 DCM, 保证不管该设备置于何处, 软件都是可用的。在另一个补充实施例中, 特定设备的 DCM 从设备制造商或因特网上第三方获得, 或者在诸如软盘的媒体设备上提供。在以上每个实施例中, DCM 56 一旦被下载, 就能被存储在各种位置。最好将 DCM 56 存储在它控制的设备上。不过, DCM 56 也能被存储在任何合适的位置。在本发明

的一个补充实施例中，用主平台支持的公用的字节或脚本代码格式，诸如 Java 或 Java 脚本语言，编写 DCM 56。DCM 56 然后被上载到主设备并在那里执行。

5 事件管理器 62 将网络内所有事件通知广播到所有感兴趣的设备。事件管理器 62 作为其节点内所有模块的中心位置，登记在该节点中发生事件时接受通知。事件管理器 62 保存一个事件通知列表数据结构，列表的内容是所定义的事件类型和所有已经登记接受每个类型的事件的通知的设备的目的地标识符。每个设备就其所感兴趣的每个事件类型向事件管理器 62 登记，提供它们的客户标识符和一个要在广播事件消息时传送回它们的令牌值 (token value)。事件是某种动作的实际发生，来自设备的消息被发往多个目的地。

10 事件管理器 62 一般不生成事件，但是用媒体管理器接收并广播由其它部件投寄的事件。在向远程节点中的客户应用程序 48 广播事件时，事件管理器 62 利用广播管理器 74。事件管理器 62 向各 DCM 56 通知正在通过控制软件级的接口发生的什么用户输入，以便各 DCM 56 能适当地处理它们的物理设备。从远程位置控制其设备的 DCM 56 将需要接收指示什么用户正在操作的消息并需要向其设备发送适当的消息。事件管理器 62 借助消息接发系统和适当定义的事件消息支持各远程 DCM 56 的执行。适当定义的事件消息包括设备管理消息（诸如在有设备添加到网络时生成的新设备消息）和用户交互消息。用户交互消息支持如下述美国临时专利申请中所描述的最佳图形用户接口。该专利申请名称为“描述基于 AV/C 的设备的人类接口特征和功能的方法”（申请号 60/054,199，申请日 1997 年 7 月 30 日），特此引用。除了适当定义的消息，任何两个 DCM 或软件模块也能定义常规或专用消息。

25 图形管理器 72 管理远程设备控件 (controls) 向控制应用程序的嵌入并支持由各 DCM 56 对自描述信息数据的远程表示。图形管理器 72 提供一个使各 DCM 56 能对屏幕空间判优并在共享的图形环境中工作的程序设计接口。这使得能通过控制软件的接口向用户表示并由用户利用设备的特定功能。

30 数据格式管理器 68 管理在设备之间流动的数据的格式。这包括能插入驻留的媒体管理器中，进行作为缓冲器管理和数据格式过程

的一部分的数据格式转换。多数数据格式转换是基于对数据的源和目的地的了解，透明地为客户应用程序进行的。其它数据换要求客户应用程序 48 建立一个格式转换过程。格式转换最好是在数据正在传送的同时直接进行。另一方面，数据格式转换也可以以数据传送的预处理任务或后处理任务的形式进行。给定平台上可用的数据格式转换服务被存储在服务注册表 59 中。除了用注册表来寻找服务，数据格式管理器 68 还负责实例化服务模块并将它们登记在服务注册表 59 中。

数据流管理器 64 与总线管理器 70 协作，提供辅助从源向目的地传送数据的服务，源与目的地之间可能包含许多节点。如果源和目的地设备使用不同的数据类型，或者隔着障碍，则数据流管理器 64 也将与数据格式管理器 68 和服务注册表 59 协作，处理自动的或所请求的数据转换服务。在等时数据的传送期间，数据流管理器 64 提供缓冲器分配和管理服务。缓冲器管理包括提供一致的通知机构以通知客户应用程序何时数据可用于处理。在等时数据流入客户应用程序 48 的同时，各种存储缓冲器中都充满数据。数据流管理器 64 通知客户应用程序每个缓冲器何时充满，以便它能处理从该缓冲器获取数据的过程。此外，因为有对捕获的数据优化的适当服务模块分区存储器，为客户应用程序简化了缓冲器管理。这包括将所分配的存储器为一个视频数据流分割成扫描行或帧大小的段或者为原始音频或 MIDI 数据分割成最佳的段大小。

图 5 表示用本发明的媒体管理器设置两个设备之间的数据传送所涉及的步骤的流程图。该方法始于方框 100。当客户应用程序 48 在两个设备之间为数据传送而建立连接时，该应用程序调用代表这两个设备的两个 DCM 56 的其中之一 25 的 EstablishExternalConnection() 过程并将另一个 DCM 56 的 moduleID 值（模块标识符值）作为参数传送。（方框 102）被调用的 DCM 56 然后调用数据流管理器 64 去帮助进行连接，并将源和目的地 ModuleID 作为参数传送。（方框 104）数据流管理器 64 然后分析源和目的地标识符，确定它们处于不同的节点。（方框 106）数据流管理器 64 30 然后从源节点的总线管理器 7 获得网络的拓扑图。（方框 108）数据流管理器 64 然后分析该拓扑图，寻找目的地节点，判断它是否在拓

扑图上。(方框 110)如果目的地节点在拓扑图上,数据流管理器 64 就跳转到方框 118,去为数据传送确定最佳路由。如果目的地节点不在拓扑图上,数据流管理器 64 就从服务注册表 59 获得目的地 DCM,以便确定该节点的传输协议。(方框 112)数据流管理器 64 然后寻找合适的传输协议服务模块并建立合适的转换过程。(方框 114)然后判断是否需要桥接多个传送。(方框 116)如果需要桥接多个传送,数据流管理器 64 就跳转回方框 114,获得另一个传送转换模块。否则,数据流管理器 64 就分析连接路径,以确定数据流的最佳路线。(方框 118)数据流管理器 64 然后分析源和目的地节点的输入数据格式,以判定是否需要转换。(方框 120)如果需要转换,数据流管理器 64 就根据输入和输出格式从服务注册表 59 获得合适的格式转换程序并建立转换过程。否则,数据流路由完成,这两个设备之间的数据传送可以开始。(方框 124)

总线管理器 70 抽象基础设备互连机构,提供一个描述总线体系结构的功能的公用的程序设计接口集。在本发明最佳实施例中,设备由 IEEE 1394-1995 串行总线连接。对于 IEEE 1394-1995 串行总线网络来说,总线管理器 70 驻留在由主机操作系统 58 提供的 IEEE 1394-1995 HAL 层的顶上。总线管理器 70 能帮助综合直到本发明的媒体管理器的总线管理活动。总线管理器 70 以通过事件管理器 62 发出总线复位通知并提供关于环境如何改变的完整信息的方式,在发生总线复位活动时通知客户应用程序 48。接收该信息的客户应用程序被提供以关于可能已突然消失的设备和在总线复位后突然已变得可用的设备的信息。

总线管理器 70 也向客户应用程序 48 提供拓扑图、速度图和其它环境描述。拓扑图中的信息被用来建立一个帮助用户明白设备的连接和如何使用某些特征功能的用户接口。该信息也被用来提供如上针对数据流管理器 64 所述的自动数据路由选择。速度图被用来分析当前的连接图,为用户通过重新安排设备的连接方式而提高网络上设备的性能提供有益的提示。总线管理器 70 也提供为两个节点或节点内的软件模块提供原子级 (atomic-level) 的数据通信服务,以便以最佳格式或协议彼此发送字节。这个协议是在这些原子通信功能的顶部建立的。

在总线的总线复位或变化通知之后，总线管理器 70 向所有刚出现的设备分配新的 ID 值并确定哪些设备已经消失。总线管理器 70 然后调用 DCM 管理器 54 去为刚出现的设备创建新的 DCM 56，并将总线变化通知邮寄到事件管理器 62，它将通知所有注册客户关于总线复位的消息。这个通知提供足够的信息供客户应用程序 48 确定总线上哪些设备已经改变。

传送适配模块 78 负责组装消息数据，然后将其传递给 HAL 供实际传送到目的地设备。HAL 位于本发明的媒体管理器的最低层。该层提供一个公用的程序设计接口，向上连接诸如各 DCM 56 的客户 (clients) 和任何其它需要与其通信的实体。传送适配模块 78 使用如上所述的总线管理器 70 的原子消息接发功能。

如上所述，各 DCM 56 通过输出用于归客户应用程序 48 负责的设备控制的标准化接口，提供一个协议抽象服务。由各 DCM 56 为设备控制提供的程序设计接口被划分成公用的音像控制级和特定于设备的音像控制级。公用的音像控制级为包括基本传送控制功能的公用命令，诸如播放、停止、快进、倒退命令，提供一个接口。特定于设备的控制级提供一个接口用于包括对给定类型的设备通用的特征功能—诸如对于具有录制功能的设备来说的录制命令，和特定于一定型号或某组设备的特征功能。各 DCM 56 所提供的协议抽象服务保证各模块与应用程序 48 之间的程序设计接口总是相同的，不管使用的什么种类的设备和控制协议。这个特点使得应用程序和用户能有很大的灵活性。各 DCM 56 也提供一个用户输入事件抽象模型，使得客户应用程序能在用户与图形用户接口元件交互时显示图形用户接口元件并向 DCM 56 发送标准用户事件消息，正如上文提及的美国临时专利申请（申请号 60/054,199）中所描述的那样。

本发明的数据流管理器提供数据流管理和其它服务。媒体管理器作为主机操作系统 48 的扩展，向媒体管理器平台的其它部件以及向客户应用程序 48 提供各种服务。媒体管理器管理并组织各 DCM 56。媒体管理器发现并初始化适合于现存应用程序的 DCM 56，同时去除不需要的 DCM 56。在系统每次启动时，或者当例如因 IEEE 1394-1995 复位，系统可能变化的任何时候时，媒体管理器遵循特定的顺序。媒体管理器也提供一个围绕主机操作系统上使用的特定动态链接库

解决方案的包装(wrapper)。这使得能在给定的操作系统 58 上用最佳动态链接库来实现模块,与此同时对外部应用程序仍然保持一个一致的接口。

5 媒体管理器也负责管理网络上设备之间数据传送操作的流和格式。在管理数据流时,媒体管理器将以独立于正在使用的操作系统的方式分配和管理适当的缓冲器。

媒体管理器也提供 IEEE 1394-1995 总线环境的高级协议管理。为了全面支持动态的设备动作,正如至用户级的热闭塞(hot plugging up),应用程序和设备需要知道对 IEEE 1394-1995 总线环境的改变。
10 媒体管理器负责通过总线管理器 70 和事件管理器 62,以发出总线复位通知和提供关于环境已经如何变化的完整信息的方式,通知应用程序和设备在 IEEE 1394-1995 总线上已经发生了总线复位活动。媒体管理器也通过总线管理器 70 向应用程序和设备提供拓扑图和其它环境描述。拓扑图描述 IEEE 1394-1995 网络内设备之间的连接。从
15 拓扑图导出的信息被用来建立一个帮助用户了解设备之间如何连接和如何使用某些特征功能的人界面。

应用程序服务模块 60 提供一个在主机操作系统 58 和应用程序 48 之间的服务层,以便为应用程序提供独立于正被使用的操作系统的基本功能。该功能包括提供存储器分配和比大多数操作系统中可用的基本功能更坚固的处理例程,和提供设备配置和控制模块—这些
20 模块是自包容的独立模块,被调用时提供所有的用户接口和交互管理。

传送适配模块 78 提供一个连接设备控制模块 50 和连接应用程序 48 的公用程序设计接口,注意通过主机操作系统 58 提升协议功能。系统级接口块 50 的内部设计和实现,利用所使用的特定主机操
25 作系统的体系结构,以便实现应用程序 48 可用的 IEEE 1394-1995 功能。

本发明的媒体管理器平台包括各 DCM 56、应用程序服务模块 60 和用于传送适配模块 78 所提供 IEEE 1394-1995 总线协议的系统级
30 接口。在普通操作过程中,应用程序 48 将与所有这些部件通信。在与 DCM 56 通信时,程序 48 将使用单一的程序设计接口。在与应用程序服务模块 60 通信时,程序 48 也将使用单一的程序设计接口。

图 3 和 4 中表示的如上所述的客户应用程序 48 是一个就本发明的媒体管理器平台的体系结构而言存在于所有其它部件之上的实体。为了完成要求其完成的大部分任务, 应用程序 48 将通过本地信使与各 DCM 56 和现有的各应用程序服务模块 60 通信。必要的时候, 应用程序 48 通过主机操作系统 58 使用该体系结构的较低的各层。

当客户应用程序 48 启动时, 客户应用程序 48 必须初始化并向媒体管理器登记。客户应用程序 48 初始化媒体管理器是为了确信媒体管理器处于启动和运行状态, 准备好服务应用程序 48。客户应用程序 48 向媒体管理器登记是为了向媒体管理器提供为与应用程序 48 交互而必需的所有信息, 并向消息接发系统登记。应用程序 48 在启动时, 一般要确信主机操作系统已经被初始化, 有最低限度的服务可用, 有可用于运行的必要的内存空间。这些步骤是在应用程序 48 初始化媒体管理器后由媒体管理器为应用程序执行的。

客户应用程序启动时, 遵循图 6 的流程图中所示的步骤。应用程序 48 在步骤 140 启动。启动后, 应用程序 48 初始化媒体管理器。在本发明最佳实施例中, 应用程序通过下列调用初始化媒体管理器:

```
err=SMM.Initialize
```

媒体管理器在初始化时, 将分配必要的内存和系统服务来支持应用程序 48。

媒体管理器的初始化完成之后, 应用程序于是在步骤 144 向媒体管理器登记。这个登记步骤使应用程序 48 能向媒体管理器提供媒体管理器为适当地支持应用程序 48 而需要的特定信息。例如, 应用程序 48 必须提供一个回调例程的地址, 该例程用于通知有关环境的重要事件, 包括 IEEE 1394-1995 总线复位、异步事务处理完成和存储缓冲器充满一定数量等时数据时的触发器。登记步骤由下列指令完成:

```
SonyErrorResultTypeSMM_RegisterClient(SMMClientIdentifierType*theClientID,  
SMMBusEventNotificationUPP clientBusEventNotificatinCallback,  
void*clientBusEventCallbackData);
```

参数 theClientID 是媒体管理器为应用程序创建的独有标识符。将来在与媒体管理器通信时, 例如当要关闭并向媒体管理器注销 (unregistering) 时, 要求应用程序 48 将该标识符传递回来。参数 clientBusEventNotificatinCallback 是个对应用程序将要执行

的回调函数的适当格式化的引用。如果应用程序 48 不需要了解网络环境可能发生的动态变化, 则不要求应用程序 48 执行这种回调函数。如果应用程序 48 不执行这个回调函数, 应用程序将为该参数传递一个 NIL 值。

5 参数 clientBusEventCallbackData 可以是应用程序将要在回调例程中要求使用的任何值。该值一般是个指向一个存储块的指针, 当媒体管理器调用回调例程时, 它将把该值传递回客户应用程序 48, 使应用程序 48 能访问全局存储器 (global storage) 或其它适当的数据。

10 为了完成向媒体管理器登记的步骤, 应用程序 48 也必须用下列接口执行通知回调函数:

```
pascal void(*SMMBusEventNotificationProcPtr)(void*clientData, SMMBusEventType
busEventIndicator, SMMBusEventRecPtr busEventInfo);
```

15 参数 clientData 是传入到登记功能的 clientBusEventCallbackData 参数。参数 busEventIndicator 是枚举数据类型的, 它指出应用程序被通知的是何种类型的事件。规定的事件包括执行复位、何时设备插入网络或从网络拔出、异步事务处理的完成以及指定缓冲器在等时数据传送期间何时充满。参数 busEventInfo 提供含有特定事件的相关信息的数据结构。

20 在完成向媒体管理器登记的步骤之后, 应用程序 48 将在步骤 146 获得可用的各 DCM 56。通过获得可用的各 DCM 56, 应用程序 48 将知道在网络内连接的其它类型的设备。该步骤由一系列子步骤组成。用一个循环的回调模型作为向应用程序 48 传送数据的数据传送方法。应用程序 48 然后进入一个循环, 重复的从媒体管理器请求关于
25 下一个模块的信息, 直到没有剩下的 DCM。媒体管理器一个有必要信息的数据结构, 通过回调函数将其传送给应用程序 48。接收到关于每个特定 DCM 56 的信息后, 应用程序 48 就复制其需要的信息。这个过程一直重复到所有可用的 DCM 56 都被应用程序 48 接收为止。在本发明另一种形式的实施例中, 客户应用程序查询注册表,
30 请求处理每一个可用的 DCM 56。

应用程序 48 为获得可用的 DCM 56 所必须执行的最佳回调函数定义如下:

```
void DeviceInfoCallbckRoutine(void*userData, SMMDeviceIndexType deviceIndex,
```

SonyAvDeviceRecPtr deviceInfo)

该回调函数的参数 `userData` 是媒体管理器与应用程序 48 之间数据传送的工具。应用程序 48 将定义其自己的数据结构，为这些结构的其中之一分配内存并将该结构的地址传递给媒体管理器。该地址然后被传递回这个回调函数中，使应用程序 48 能为了向其中复制信息而使用该数据结构。

该回调函数的参数 `deviceIndex` 是应用程序 48 为获得关于可用的 DCM 56 的信息而进入的循环的下标值。在回调函数中，该参数被传递回应用程序 48，以便应用程序 48 将该参数与传递到回调函数中的其它信息一起保存起来。应用程序 48 在查询特定 DCM 56 而对媒体管理器进行的其它调用中，该下标值是有用的。此外，在有设备被从网络拔出或拆接后向应用程序 48 通知有设备消失时，将使用该下标值。应用程序 48 将在其专用数据结构中的专用域中为每个 DCM 56 存储这个下标值。

该回调函数的参数 `deviceInfo` 是一个指向标签为 `SonyAVDeviceRec` 的数据结构的指针，媒体管理器在其中为应用程序 48 的检索存储着各 DCM 56。该数据结构的格式，应用程序 48 和媒体管理器都知道。一旦有 DCM 56 被存储到这个数据结构内，应用程序 48 将从该数据结构中把适当的信息复制到其自己的专用数据结构中。数据结构 `SonyAVDeviceRec` 在以下的表一中定义：

表一

```
typedef struct SonyAVDeviceRec
{
    unsigned long      deviceID;          //SMMDeviceIDType?
    unsigned long      busGeneration;
    SONY_DeviceModuleRefType controlModuleReference;
    unsigned long      reserved1;
    unsigned long      reserved2;
} SonyAVDeviceRec, * SonyAVDeviceRecPtr, ** SonyAVDeviceRecHdl;
```

参数 `deviceID` 是一个 DCM 56 的标识符，相应地是一个设备的标识符。每当应用程序 48 要与一个 DCM 56 通信时，或者当应用程序 48 向媒体管理器对特定设备请求服务时，应用程序 48 使用这个标识符。

参数 `busGeneration` 是一个在每个总线复位动作之后改变的值。在每个总线复位之后，当设备被添加或去除时，关于该总线和相连

接的设备的某些信息将改变。IEEE 1394-1995 总线每次复位时，参数 busGeneration 的值都被更新。

5 参数 controlModuleReference 是一个对与指定设备相关联的 DCM 56 的引用。当应用程序要求媒体管理器代表其与该模块交易时，该引用被使用。

应用程序 48 然后将用下列函数调用请求媒体管理器生成一个可用 DCM 56 的列表和该列表内模块的数目：

10 SonyErrorResultType SMM_FindDeviceControlModules (SMMDeviceListRefType* theDeviceList, unsigned long deviceAttributes, short*numAVDevice)

参数 theDeviceList 包括可用 DCM 56 的列表被存储的位置的地址，是由媒体管理器生成并返回的。应用程序将声明一个这个类型的局部变量并将这个变量的地址传递给这个函数。

15 参数 deviceAttributes 包括应当被返回的、应用程序 48 用来规定各 DCM 56 的类型的一组按位的标志。例如，应用程序 48 可能只希望连接与网络连接的活动的设备。当规定了某些标志值时，媒体管理器将筛选该列表，只留下满足条件的设备，然后将列表返回给应用程序 48。应用程序 48 可规定该列表包含的是：所有可标识的设备；只有处于启动和运行状态的设备；只有被插入，但是电源开关被关闭的设备；或只有空闲的（snoozing）设备。

参数 numAVDevice 包括在向应用程序 48 返回的列表中的 DCM 56 的数目。应用程序 48 用这个数目作为获得 DCM 56 的循环的上界。

25 应用程序 48 准备回调函数地址，然后进入循环，重复地调用媒体管理器，直到获得了关于列表内所有 DCM 56 的信息为止。在每经历一次循环，应用程序 48 就对下列函数作一次调用：

30 pascal SonyErrorResultType SMM_GetDeviceControlModuleInfo (SMMDeviceListRefType theDeviceList, SMMDeviceIndexType whichDevice, unsigned long reserved, SMMDeviceControlModuleIteratroUPP theDeviceListCallbackFunction, void*userData)

参数 theDeviceList 是从函数调用 FindAllDeviceModules () 返回的列表引用。参数 whichDevice 指明应用程序 48 正在请求的是关于哪些 DCM 56 的信息。参数 theDeviceListCallbackFunction 包括

所准备的回调函数地址。参数 `userData` 是一个对一个应用程序定义的数据结构的引用。这个引用在回调例程中被传递回应用程序 48，应用程序 48 然后将把任何需要的信息从媒体管理器传送到该数据结构。

5 为获得可用的 DCM 56 而进行的步骤的最佳完整序列，在下面的表二列出：

表二

```

SMMDeviceListRefType theDeviceList=NULL;

10
    if (nil!=theDeviceList)
        err=SMM.FindAllDeviceControlModules (&theDeviceList, kActiveDevices+kInactiveDevi
            ces, &gNumAVDevices);

15
    if (noErr==err)
    {
        gAVDeviceList=NewHandle (0);
        //为媒体管理器准备回调函数
        theDeviceInfoCallback=NewSMMDeviceControlModuleIteratorProc (DeviceInfoCalibackR
20
            outline);
        if ((nil!=theDeviceInfoCallback) && (nil!=gAVDeviceList))
        {
            for (loop=0; loop<gNumAVDevices; loop++)
            {
25
                err=SMM.GetDeviceControlModuleInfo (theDeviceList, loop, 0,
                    theDeviceInfoCallback.gAVDeviceList);

            }
            DisposeRoutineDescriptor (theDeviceInfoCallback);
        }
30
    else
        err=-1
    }

    void DeviceInfoCallbackRoutine (void*userData, SMMDeviceIndexType
        deviceindex, SonyAVDeviceRECPtrdeviceInfo)
35
    {
        //将我关心的任何信息从 deviceInfo 数据结构拷贝
        //至由 userData 引用的我的私有数据中
        (myPrivateRecordPtr)userData->deviceId=deviceInfo->deviceId;
    }

```

40 应用程序 48 获得可用的各 DCM 56 后，应用程序 48 然后将在步

5 骤 68 获得特定于设备的信息。由媒体管理器返回的 DCM 信息是系统级的信息，包括每个设备的独有标识符和特定于协议的信息—诸如 IEEE 1394-1995 设备的总线生成。为了获得特定于设备的信息—诸如设备的状态、描述性名字串和图象，应用程序 48 必须通过合适的 DCM 56 与该设备通信。通过完成图 6 中所示的上述步骤，应用程序 48 就将完成了其启动例程，处于做好运行准备的状态。

应用程序 48 在运行期间，将处理包括接收控制输入的用户和系统级的事件和消息，以及来自其它过程、主机操作系统和媒体管理器的消息。

10 以上就特定的详细实施例说明了本发明，以便于理解本发明的构成和操作的原理。本文中这种对特定实施例及其细节的引述不是要限制后附的权利要求书的范围。对本领域的熟练人员来说，显然可以对为说明而选择的实施例作出改动，而不偏离本发明的精神和范围。特别地，对本领域的技术人员来说显而易见的是，尽管本发
15 明最佳实施例用来管理在 IEEE 1394-1995 串行总线体系结构中连接在一起的设备，本发明也可以实现用来管理其它总线结构内的设备。

说明书附图

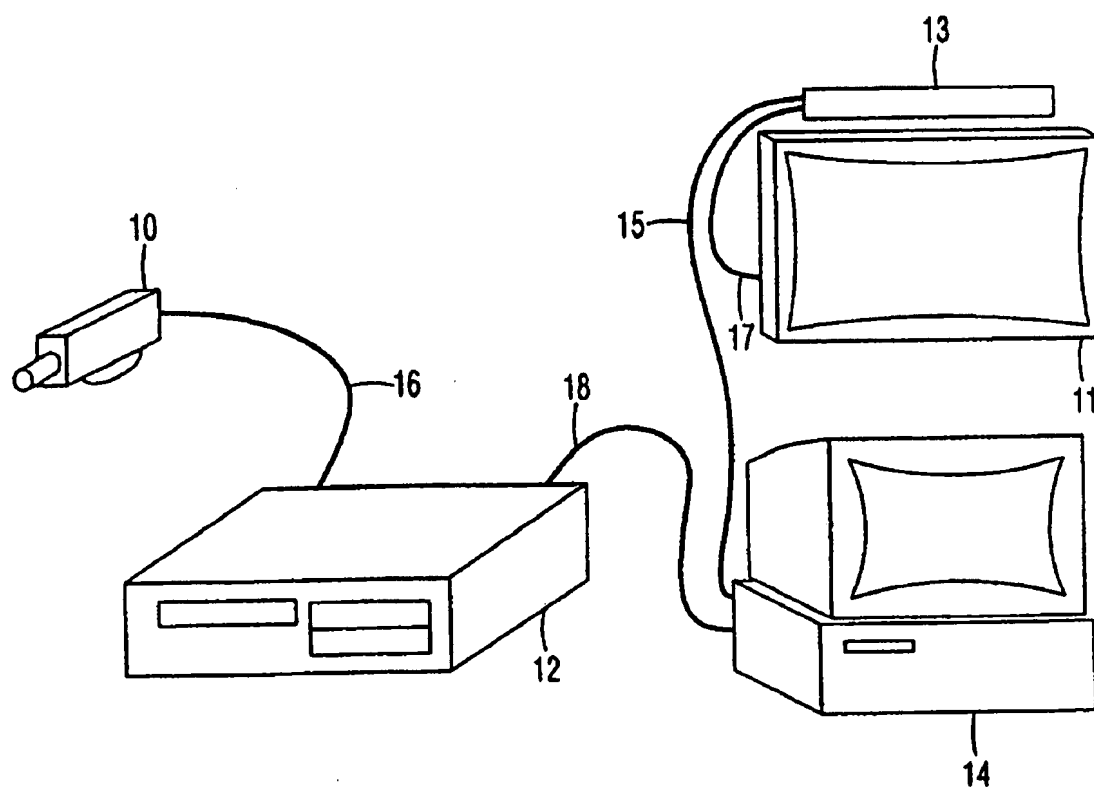


图 1

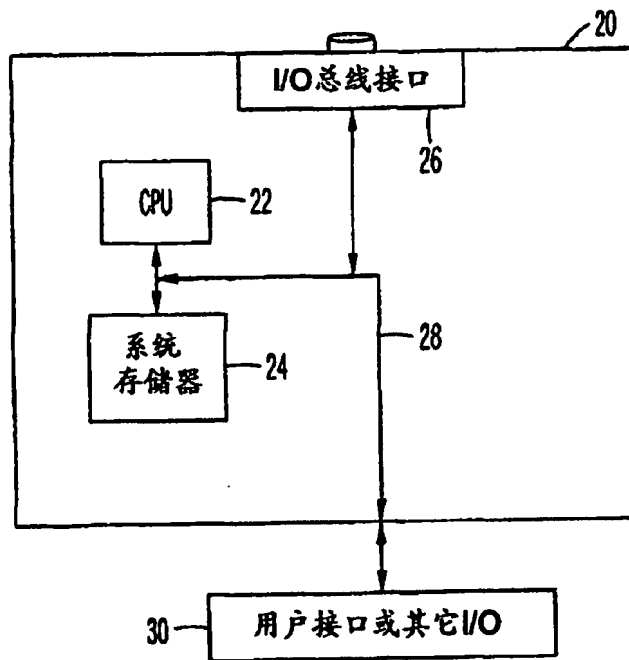


图 2

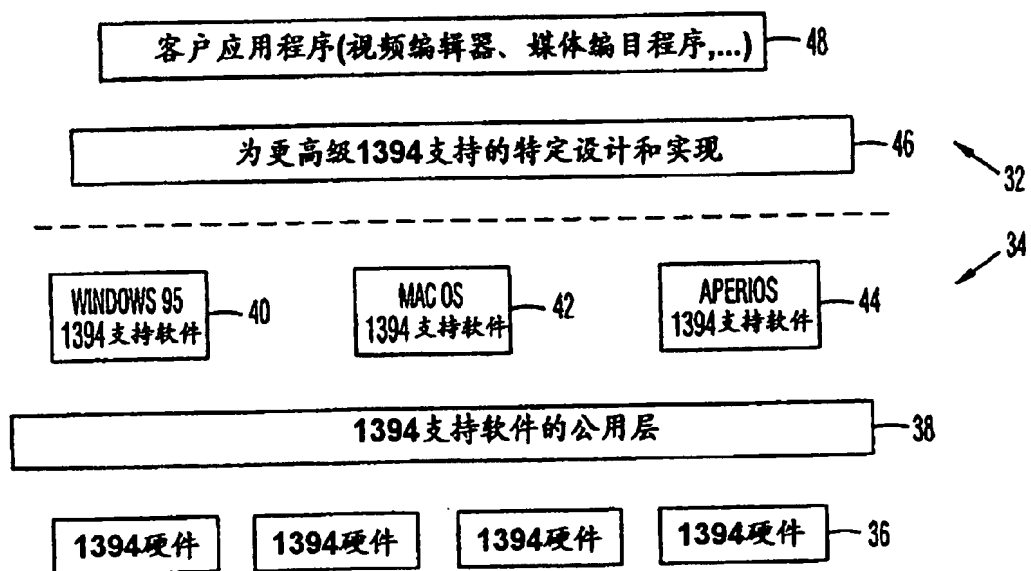


图 3

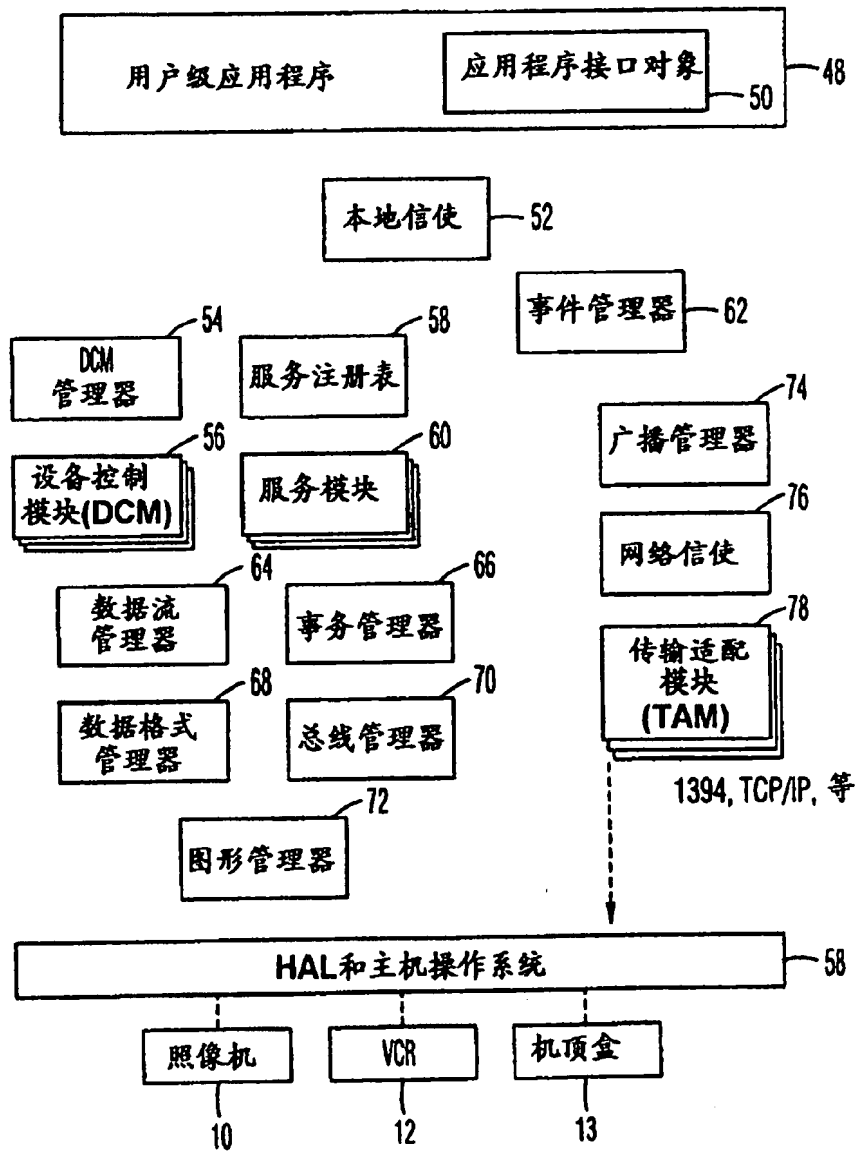


图 4

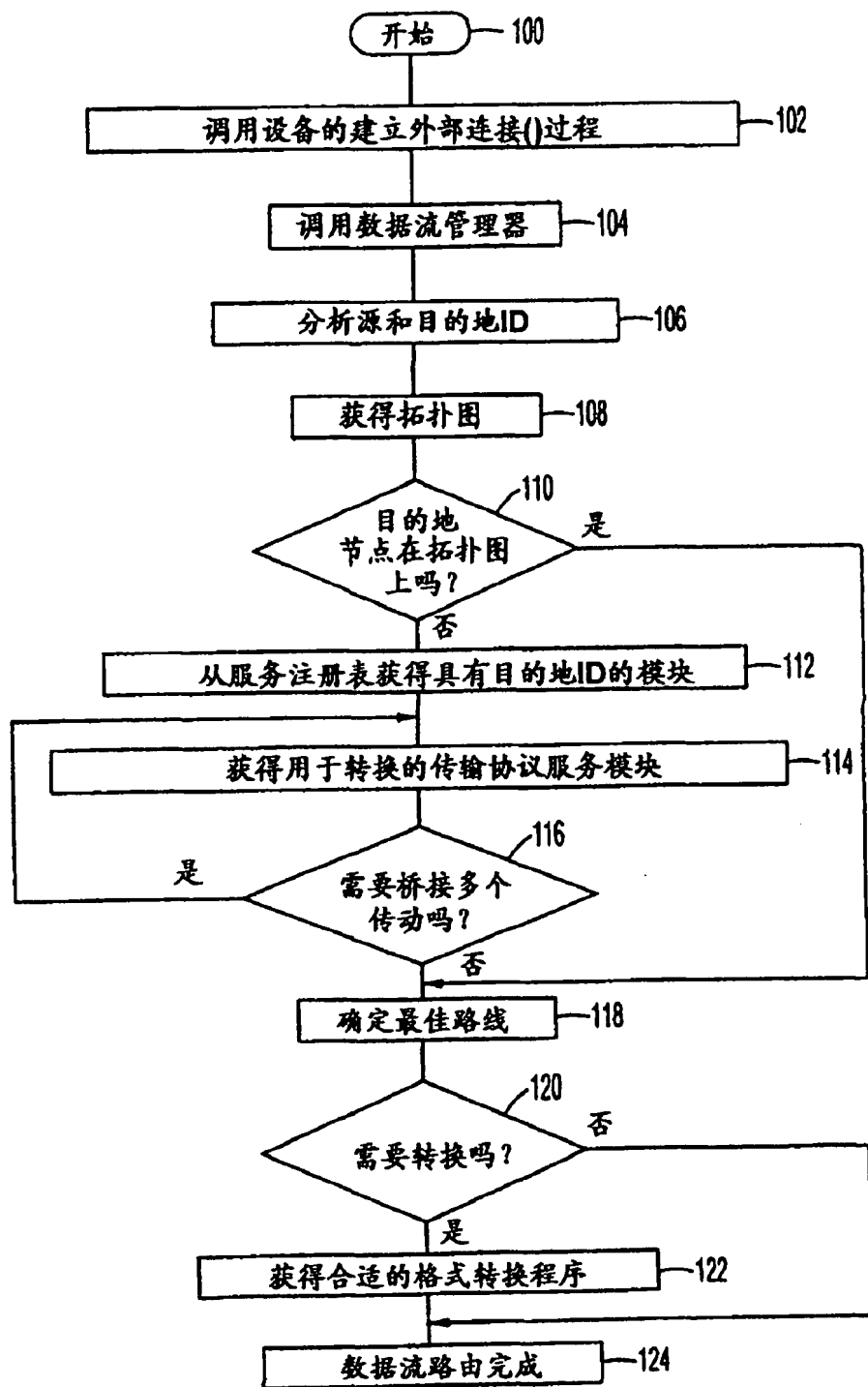


图 5

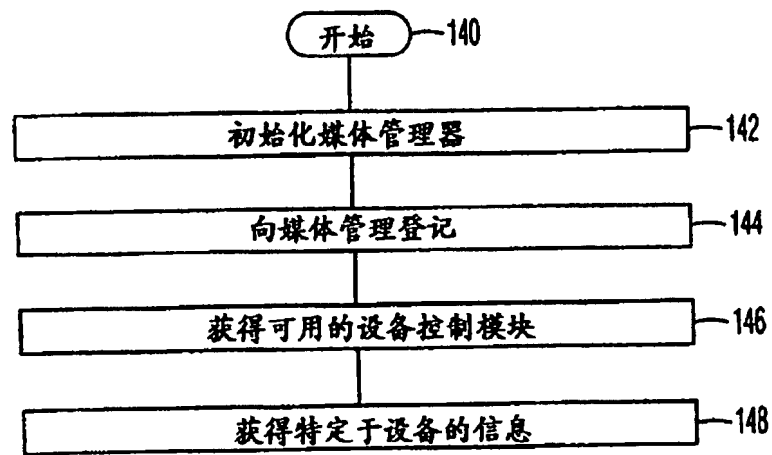


图 6